In the Claims

Claims 1, 19, 37 and 55 have been amended.

Claim 43 has been cancelled.

1. (Currently Amended) In a distributed computing environment, a computer-implemented method for implementing workflow responsive to a directory object state change, the method comprising:

automatically detecting a state change to an object in a directory, the directory corresponding to a directory schema, the directory schema defining a hierarchy of content classes, wherein at least one content class of the hierarchy includes a flexible attribute; and

responsive to detecting the state change, automatically:

mapping the state change to the object to a workflow comprising a set of tasks; and

executing the tasks to achieve a desired state in the directory.

2. (Previously Presented) The method of claim 1, wherein executing the tasks further comprises storing the desired state.

3. (Previously Presented) The method of claim 1, wherein executing the tasks further comprises continuously executing an operation of a task of the tasks until convergence of the desired state is identified.

4. (Previously Presented) The method of claim 1, wherein executing the tasks further comprises storing a sequence of operations based on the tasks.

5. (Previously Presented) The method of claim 1, wherein executing the tasks further comprises storing information corresponding to one or more directory objects to which the workflow applies.

6. (Previously Presented) The method of claim 1, wherein executing the tasks further comprises storing status information based on respective status of at least one subset of the tasks.

7. (Previously Presented) The method of claim 1, wherein mapping the state change to the object further comprises evaluating the state change to the object based on a declarative condition stored as a text string on an object instance of a content class defined by the directory schema.

8. (Previously Presented) The method of claim 1, wherein a task of the tasks comprises any combination of a declarative condition or an operation that is stored as a text string on an object instance of a content class defined by the directory schema.

9. (Previously Presented) The method of claim 1, wherein semantics of the workflow are based on a workflow schema.

10. (Previously Presented) The method of claim 1, wherein mapping the state change to the object, semantics of the mapping are based on an event association object schema.

11. (Previously Presented) The method of claim 1, wherein executing the tasks at least one subset of the tasks are executed with respect to one another based on an order of execution relationship comprising a finish-start relationship, a parallel execution relationship, a precedence constraint relationship, or a task priority relationship.

12. (Previously Presented) The method of claim 1, wherein executing the tasks at least one subset of the tasks is executed with respect to one another based on a precedence constraint relationship or a task priority relationship.

13. (Previously Presented) The method of claim 1, wherein the method further comprises:

monitoring a status corresponding to a task of the tasks;

storing the status on a status monitoring object; and

wherein a content class in the directory schema defines the status-monitoring object.

14. (Previously Presented) The method of claim 1, wherein the method further comprises:

monitoring a set of directory resources affected by the workflow;

storing the directory resources on a status monitoring object; and

wherein a content class in the directory schema defines the status-monitoring object.

15. (Previously Presented) The method of claim 1, wherein the method further comprises:

monitoring a status corresponding to an operation of the workflow;

determining that the status comprises a failure status;

responsive to the determining, taking a corrective action to advance the workflow in view of the failure status; and

wherein a content class in the directory schema defines the status-monitoring object.

16. (Previously Presented) The method of claim 1, wherein executing the tasks further comprises:

updating a status corresponding to a task in the workflow; and

responsive to the updating, evaluating the workflow to determine that a next task of the tasks to be implemented.

17. (Previously Presented) The method of claim 1, wherein the tasks represent an inverse set of tasks that were previously performed as part of a different workflow.

18. (Previously Presented) The method of claim 1, wherein the tasks implement a policy with respect to one or more directory resources, and wherein mapping the state change to the object further comprises automatically determining the workflow based on the policy.

19. (Currently Amended) A computer-readable medium comprising computer-executable instructions to implement workflow responsive to a directory object state change, the computer-executable instructions comprising instructions for:

detecting a state change to an object in a directory, wherein at least one object of the directory is defined by a flexible attribute configured to store a plurality of different data types, the functionality of the at least one object varying in accordance with the data type stored on the at least one object; and

responsive to detecting the state change:

mapping the state change to the object to a workflow comprising a set of tasks; and

executing the tasks to achieve a desired state in the directory.

20. (Previously Presented) The computer-readable medium of claim 19, wherein the instructions for executing the tasks further comprise instructions for storing the desired state.

21. (Previously Presented) The computer-readable medium of claim 19, wherein the instructions for executing the tasks further comprise instructions for continuously executing an operation of a task of the tasks until convergence of the desired state is identified.

22. (Previously Presented) The computer-readable medium of claim 19, wherein the instructions for executing the tasks further comprise instructions for storing a sequence of operations based on the tasks.

23. (Previously Presented) The computer-readable medium of claim 19, wherein instructions for executing the tasks further comprise instructions for storing information corresponding to one or more directory objects to which the workflow applies.

24. (Previously Presented) The computer-readable medium of claim 19, wherein the instructions for executing the tasks further comprise instructions for storing status information based on respective status of at least one subset of the tasks.

25. (Previously Presented) The computer-readable medium of claim 19, wherein the instructions for mapping the state change to the object further comprise instructions for evaluating the state change to the object based on a declarative condition stored as a text string on an object instance of a content class defined by a directory schema.

26. (Previously Presented) The computer-readable medium of claim 19, wherein a task of the tasks comprises any combination of declarative conditions and operations that are stored as a text string on an object instance of a content class defined by a directory schema.

27. (Previously Presented) The computer-readable medium of claim 19, wherein semantics of the workflow are based on a workflow schema.

28. (Previously Presented) The computer-readable medium of claim 19, wherein the instructions for mapping the state change to the object, semantics of the mapping are based on an event association object schema.

29. (Previously Presented) The computer-readable medium of claim 19, wherein the instructions for executing the tasks, at least one subset of the tasks are executed with respect to one another based on an order of execution relationship comprising a finish-start relationship, a parallel execution relationship, a precedence constraint relationship, or a task priority relationship.

30. (Previously Presented) The computer-readable medium of claim 19, wherein the instructions for executing the tasks, at least one subset of the tasks are executed with respect to one another based on a precedence constraint relationship or a task priority relationship.

31. (Previously Presented) The computer-readable medium of claim 19, wherein the computer-executable instructions further comprise instructions for:

automatically:

monitoring a status corresponding to a task of the tasks; and

storing the status on a status monitoring object; and

wherein a content class in the directory schema defines the status-monitoring object.

32. (Previously Presented) The computer-readable medium of claim 19, wherein the computer-executable instructions further comprise instructions for:

automatically:

monitoring a set of directory resources affected by the workflow; and

storing the directory resources on a status monitoring object; and

wherein a content class in the directory schema defines the status-monitoring object.

33. (Previously Presented) The computer-readable medium of claim 19, wherein the computer-executable instructions further comprises instructions for automated operations for:

monitoring, by a status-monitoring object defined by a content class in the directory schema, a status corresponding to an operation of the workflow;

determining that the status comprises a failure status;

responsive to the determining, taking a corrective action to advance the workflow in view of the failure status

.

34. (Previously Presented) The computer-readable medium of claim 19, wherein the instructions for executing the tasks further comprise instructions for:

updating a status corresponding to a task in the workflow; and

responsive to the updating, evaluating the workflow to determine that a next task of the tasks to be implemented.

35. (Previously Presented) The computer-readable medium of claim 19, wherein the tasks represent an inverse set of tasks that were previously performed as part of a different workflow.

36. (Previously Presented) The computer-readable medium of claim 19, wherein the tasks implement a policy with respect to one or more directory resources, and wherein the instructions for mapping the state change to the object further comprises instructions for automatically determining the workflow based on the policy.

37. (Currently Amended) A computing device comprising:

a memory comprising computer-executable instructions for automatically implementing workflow responsive to a directory object state change; and

a processor coupled to the memory for executing the computer-executable instructions, the computer-executable instructions comprising instructions for:

detecting a state change to an object in a directory, the directory corresponding to a directory schema; and

responsive to detecting the state change:

mapping the state change to the object to a workflow comprising a set of tasks, the mapping including evaluating the state change to the object based on a declarative condition stored as a text string on an object instance of a content class defined by the directory schema; and

executing the tasks to achieve a desired state in the directory.

38. (Previously Presented) The computing device of claim 37, wherein the instructions for executing the tasks further comprise instructions for storing the desired state.

39. (Previously Presented) The computing device of claim 37, wherein the instructions for executing the tasks further comprise instructions for continuously executing an operation of a task of the tasks until convergence of the desired state is identified.

40. (Previously Presented) The computing device of claim 37, wherein the instructions for executing the tasks further comprise instructions for storing a sequence of operations based on the tasks.

41. (Previously Presented) The computing device of claim 37, wherein instructions for executing the tasks further comprise instructions for storing information corresponding to one or more directory objects to which the workflow applies.

42. (Previously Presented) The computing device of claim 37, wherein the instructions for executing the tasks further comprise instructions for storing status information based on respective status of at least one subset of the tasks.

43. (Cancelled)

44. (Previously Presented) The computing device of claim 37, wherein a task of the tasks comprises any combination of one or more declarative conditions and one or more operations represented by a text string stored on an object instance of a content class defined by the directory schema.

45. (Previously Presented) The computing device of claim 37, wherein semantics of the workflow are based on a workflow schema.

46. (Previously Presented) The computing device of claim 37, wherein the instructions for mapping the state change to the object, semantics of the mapping are based on an event association object schema.

47. (Previously Presented) The computing device of claim 37, wherein the instructions for executing the tasks, at least one subset of the tasks are executed with respect to one another based on an order of execution relationship comprising a finish-start relationship, a parallel execution relationship, a precedence constraint relationship, or a task priority relationship.

48. (Previously Presented) The computing device of claim 37, wherein the instructions for executing the tasks, at least one subset of the tasks are executed with respect to one another based on a precedence constraint relationship or a task priority relationship.

49. (Previously Presented) The computing device of claim 37, wherein the computer-executable instructions further comprise instructions for:

monitoring a status corresponding to a task of the tasks;

storing the status on a status monitoring object; and

wherein a content class in the directory schema defines the status-monitoring object.

50. (Previously Presented) The computing device of claim 37, wherein the computer-executable instructions further comprise instructions for:

monitoring a set of directory resources affected by the workflow;

storing the directory resources on a status monitoring object; and

wherein a content class in the directory schema defines the status-monitoring object.

51. (Previously Presented) The computing device of claim 37, wherein the computer-executable instructions further comprises instructions for:

monitoring a status corresponding to an operation of the workflow;

determining that the status comprises a failure status;

responsive to the determining, taking a corrective action to advance the workflow in view of the failure status; and

wherein a content class in the directory schema defines the status-monitoring object.

52. (Previously Presented) The computing device of claim 37, wherein the instructions for executing the tasks further comprise instructions for:

updating a status corresponding to a task in the workflow; and

responsive to the updating, evaluating the workflow to determine that a next task of the tasks to be implemented.

53. (Previously Presented) The computing device of claim 37, wherein the tasks represent an inverse set of tasks that were previously performed as part of a different workflow.

54. (Previously Presented) The computing device of 37, wherein the tasks implement a policy with respect to one or more directory resources, and wherein the instructions for mapping the state change to the object further comprises instructions for automatically determining the workflow based on the policy.

55. (Currently Amended) A computing device comprising automated processing means for:

detecting a state change to an object in a directory, the directory corresponding to at least one content class, wherein at least one object of the at least one content class is defined by a flexible attribute, and wherein the functionality of the at least one object varies in accordance with the data type stored thereon; and

responsive to detecting the state change:

mapping the state change to the object to a workflow comprising a set of tasks; and

executing the tasks to achieve a desired state in the directory.

56. (Previously Presented) A computing device of claim 55, wherein the means for executing the tasks further comprise means for storing the desired state.

57. (Previously Presented) A computing device of claim 55, wherein the means for executing the tasks further comprise means for continuously executing an operation of a task of the tasks until convergence of the desired state is identified.

58. (Previously Presented) A computing device of claim 55, wherein the means for executing the tasks further comprise means for storing a sequence of operations based on the tasks.

59. (Previously Presented) A computing device of claim 55, wherein means for executing the tasks further comprise means for storing information corresponding to one or more directory objects to which the workflow applies.

60. (Previously Presented) A computing device of claim 55, wherein the means for executing the tasks further comprise means for storing status information based on respective status of at least one subset of the tasks.

61. (Previously Presented) A computing device of claim 55, wherein the means for mapping the state change to the object further comprise means for evaluating the state change to the object based on a declarative condition stored as a text string on an object instance of a content class defined by the directory schema.

62. (Previously Presented) A computing device of claim 55, wherein a task of the tasks comprises any combination of one or more declarative conditions and one or more operations represented by a text string stored on an object instance of a content class defined by the directory schema.

63. (Previously Presented) A computing device of claim 55, wherein semantics of the workflow are based on a workflow schema.

64. (Previously Presented) A computing device of claim 55, wherein the means for mapping the state change to the object, semantics of the mapping are based on an event association object schema.

65. (Previously Presented) A computing device of claim 55, wherein the means for executing the tasks, at least one subset of the tasks are executed with respect to one another based on an order of execution relationship comprising a finish-start relationship, a parallel execution relationship, a precedence constraint relationship, or a task priority relationship.

66. (Previously Presented) A computing device of claim 55, wherein the means for executing the tasks, at least one subset of the tasks are executed with respect to one another based on a precedence constraint relationship or a task priority relationship.

67. (Previously Presented) A computing device of claim 55, further comprising processing means for:

monitoring a status corresponding to a task of the tasks;

storing the status on a status monitoring object; and

wherein a content class in the directory schema defines the status-monitoring object.

68. (Previously Presented) A computing device of claim 55, further comprising automated processing means for:

monitoring a set of directory resources affected by the workflow;

storing the directory resources on a status monitoring object; and

wherein a content class in the directory schema defines the status-monitoring object.

69. (Previously Presented) A computing device of claim 55, further comprising automated processing means for:

monitoring a status corresponding to an operation of the workflow;

determining that the status comprises a failure status;

responsive to the determining, taking a corrective action to advance the workflow in view of the failure status.

70. (Previously Presented) A computing device of claim 55, wherein the automated processing means for executing the tasks further comprise means for:

updating a status corresponding to a task in the workflow; and

responsive to the updating, evaluating the workflow to determine that a next task of the tasks to be implemented.

71. (Previously Presented) A computing device of claim 55, wherein the tasks represent an inverse set of tasks that were previously performed as part of a different workflow.

72. (Previously Presented) A computing device of claim 55, wherein the tasks implement a policy with respect to one or more directory resources, and wherein the means for mapping the state change to the object further comprise means for automatically determining the workflow based on the policy.

73. (Previously Presented) A computer-readable medium comprising workflow enabled directory schema for automated workflow implementation by a set of computer-program instructions executable by a processor, the workflow enable directory schema comprising a plurality of base object content classes comprising:

a provisioning service content class to detect an event corresponding to a state change in a directory object;

a workflow content class for storing a sequence of tasks;

an event association content class for storing declarative conditions to map the state change to the directory object to an object instance of the workflow content class; and

wherein the provisioning service content class is further configured to execute the sequence of tasks corresponding to the object instance.

74. (Previously Presented) The computer-readable medium of claim 73, wherein at least a subset of the base object content classes comprise a respective flexible attribute data field that indicates a data type, the data type being used to express various operational or data providing properties of the flexible attribute, the various operational or data providing properties being independent of the data type and independent of any modification to the workflow enabled directory schema.

75. (Previously Presented) The computer-readable medium of claim 73, wherein the sequence of tasks comprises any combination of a declarative conditions and operations corresponding to directory-based objects.

76. (Previously Presented) The computer-readable medium of claim 73, further comprising a status monitoring content class for storing a status of an object instance of the workflow content class.

77. (Canceled)

78. (Previously Presented) A computer comprising the processor coupled to a memory comprising the computer-readable medium of claim 73.